Accelerate Performance on the Parallel Programming Super Highway

Garth Black, SSTC 2010

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
APR 2010		**00-00-2010 to 00-00-2010**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Accelerate Performance on the Paallel Programming Super Highway	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **National Instruments,11500 N Mopac Expwy,Austin,TX,78759-3504**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
Presented at the 22nd Systems and Software Technology Conference (SSTC), 26-29 April 2010, Salt Lake City, UT. Sponsored in part by the USAF. U.S. Government or Federal Rights License

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**25**	

Presentation Highlights

- Computational programming demands continue to increase at a rapid pace despite technological challenges and limitations
- Parallelism is the [new] principal method for increasing and improving processor performance
- Dataflow programming languages address several barriers associated with parallel programming
- Dataflow languages ought to be considered along with traditional (imperative) programming solutions

Email garth.black@ni.com with questions

Programming Demands and Limitations

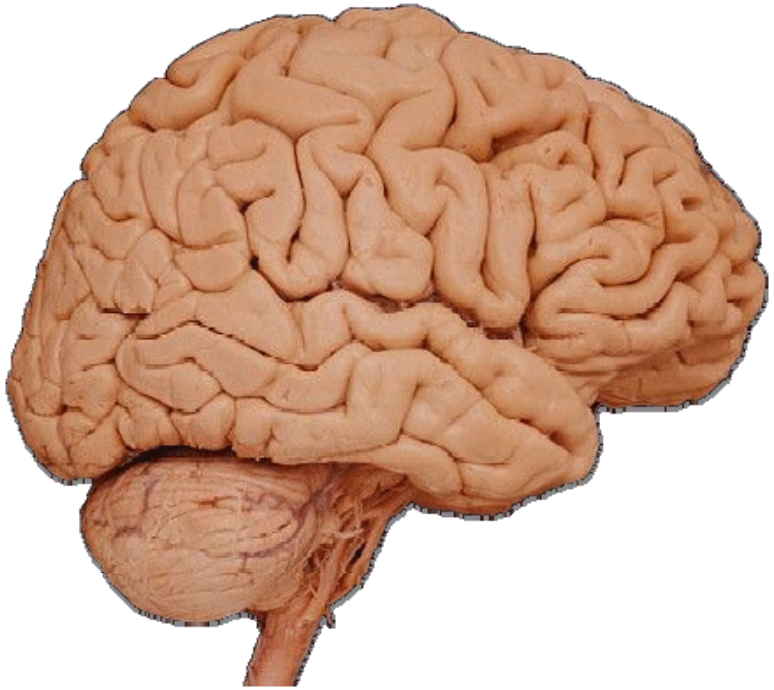
- Rising demand for faster execution and increasingly complex programming
- Clock frequency (speed) is trending to an asymptotic condition (3 GHz)
- Moore's Law may still be valid, but the Law of Thermodynamics is also valid
- Parallel Programming options exist, but can be complicated

Just increase Clock Frequency?

- Old (Conventional Wisdom)
  - Increasing clock frequency is the primary method of improving processor performance.
- New [conventional wisdom]:
  - Increasing parallelism is the primary method of improving processor performance.
- "Even representatives from Intel warned that traditional approaches to maximizing performance through maximizing clock speed have been pushed to their limit."

Asanovic, Krste et al. (December 18, 2006). "The Landscape of Parallel Computing Research: A View from Berkeley" (PDF). University of California, Berkeley. Technical Report No. UCB/EECS-2006-183.

The Human Parallel Processor



- Billions of Nerve Cells (Neurons)
- Networks of neurons form massive parallel processing system
- Parallelism: Vision, Hearing, Motion

Image courtesy of Wikipedia

"Massive" CPU Parallel Processor

- "Massively Parallel Processor"

- A cabinet from Blue Gene/L, ranked as the fourth fastest supercomputer in the world.
- More than 100 CPUs with high speed interconnect
- Analogous to Human Brain

How do we program Parallel Processes?

- Newsweek Article

(*Moore's Law Doesn't Matter*; August 15, 2009)

- Imperative vs. Dataflow programming

Imperative Programming vs. Dataflow Programmaing

- Imperative programming is modeled as a series of operations, the data paths between operations being effectively invisible
  - Examples: C/C++, Fortran, Pascal
- Dataflow programming explicitly illustrates the "flow of data" between program operations
  - Examples: SISAL, SAC, LabVIEW, VEE

Contrast: Imperative Programming vs. Dataflow Programming

Imperative Language

Line 1: x = 5

Line 2: y = x + 1

Line 3: z = y * 3

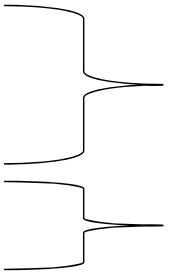
Execute each statement in order.

Dataflow Language

y * 3 -> z

x + 1 -> y

5 -> x



Rules

Input Data

Identify all rules and then provide inputs.

The compiler determines that y needs to be calculated before z.

Contrast: Imperative Programming vs. Dataflow Programming

Imperative Language

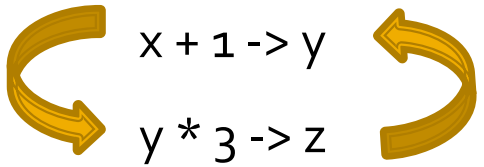
Line 1: x = 5

Line 2: y = x + 1

Line 3: z = y * 3

Execute each statement in order.

Dataflow Language



x + 1 -> y

y * 3 -> z

Rules

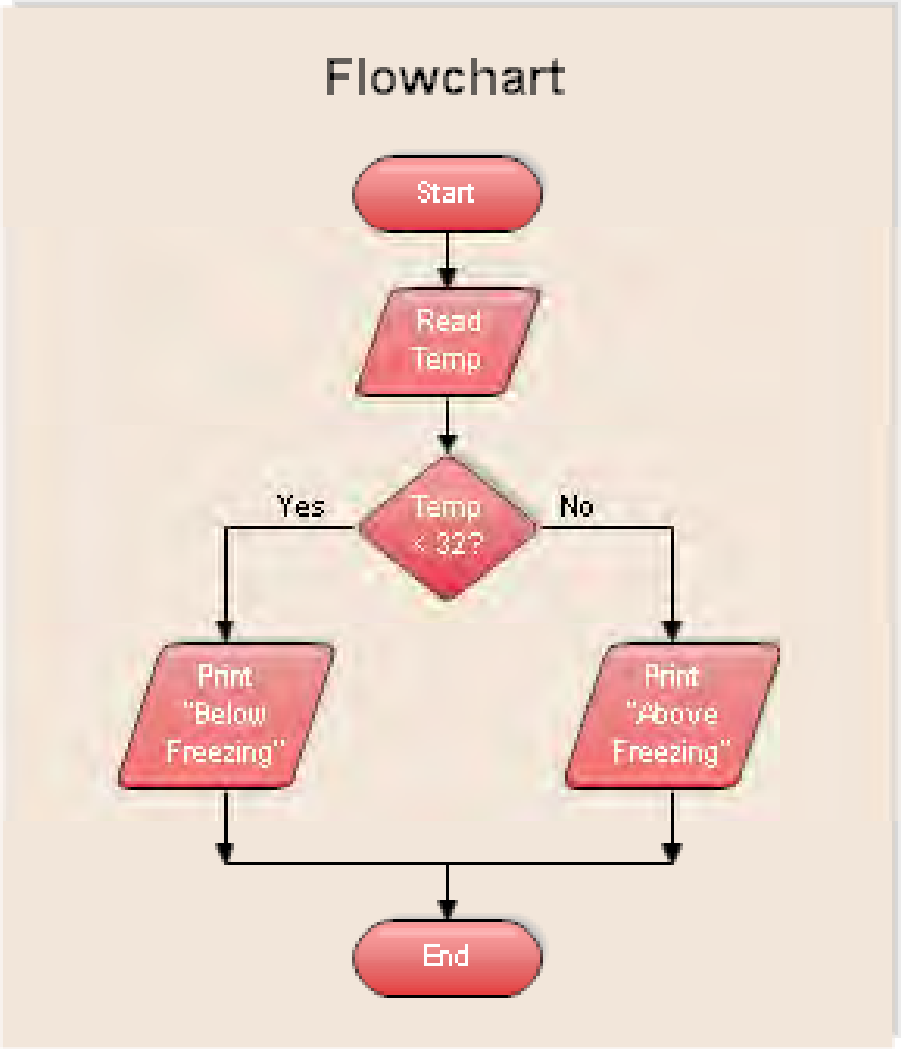
5 -> x

Input Data

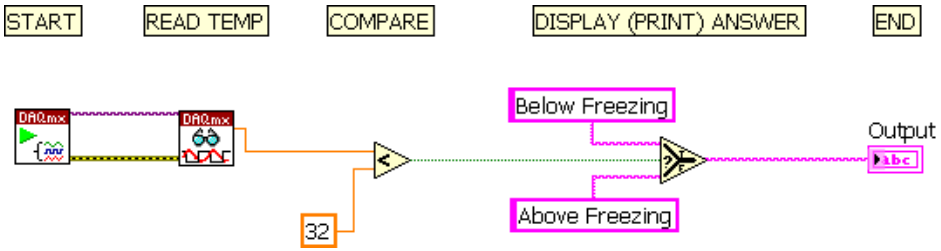
Identify all rules and then provide inputs.

The exact order of rule statements is not important in dataflow code!

Dataflow Programming correlates to standard flowchart models

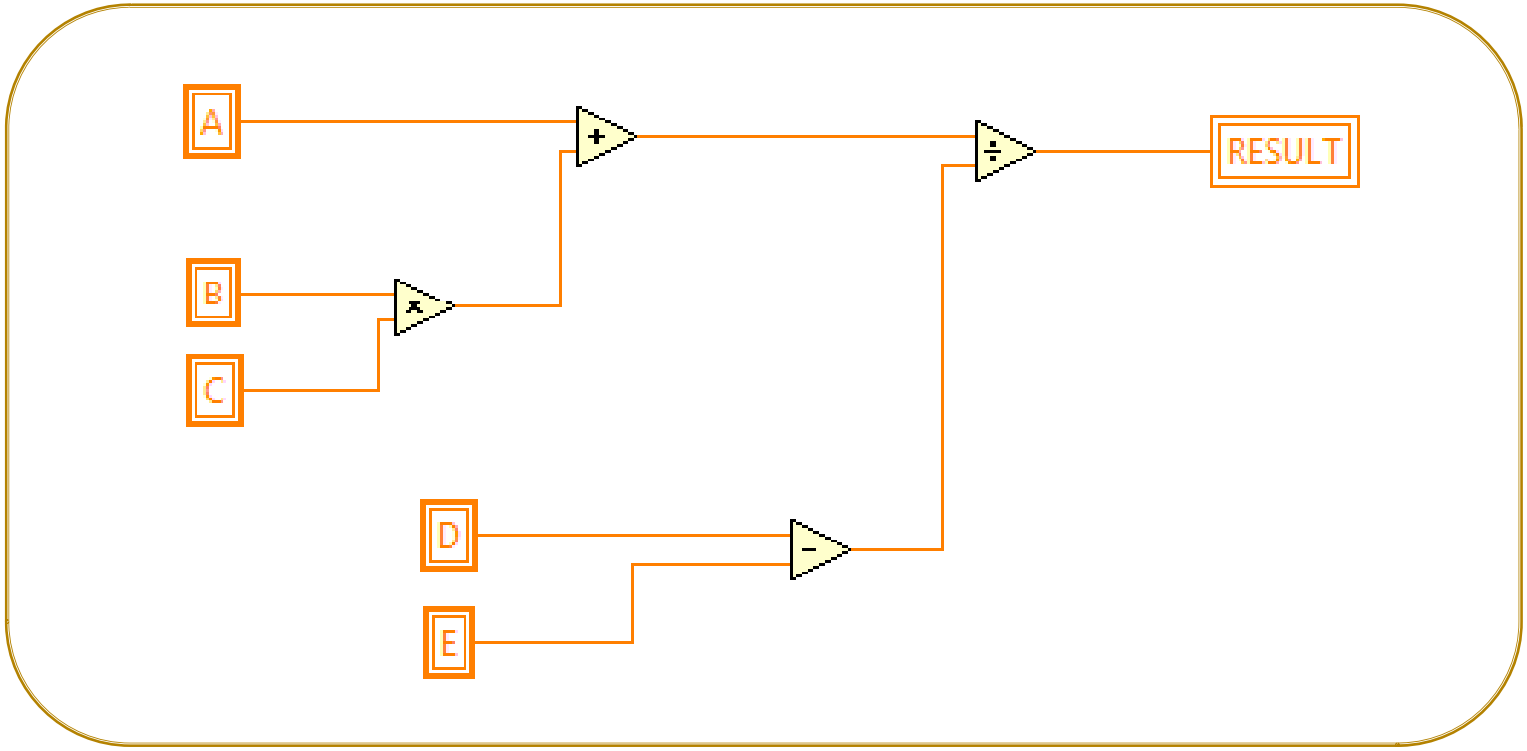


Dataflow Program



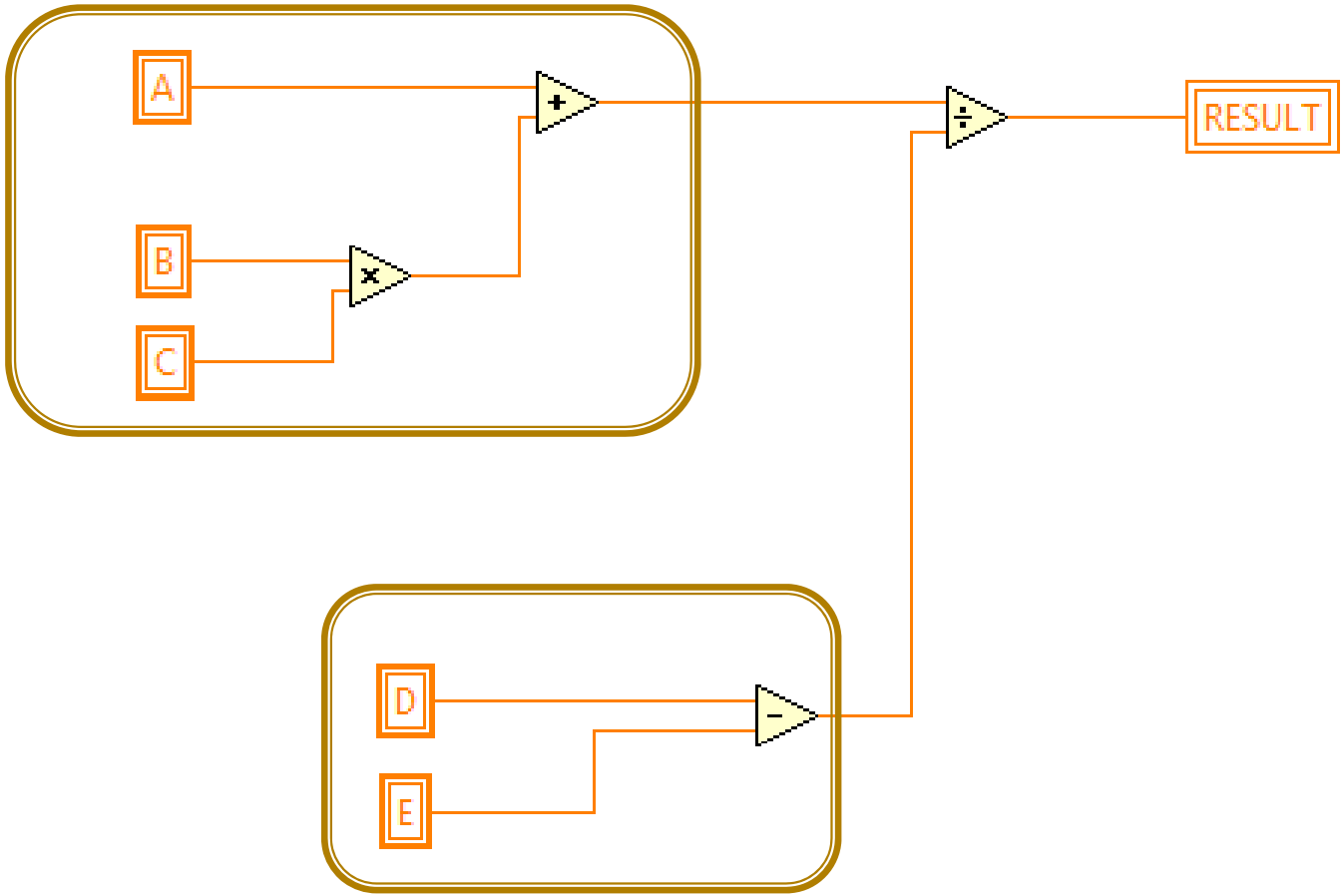
Example: A flow chart represents the relationships between inputs and outputs. Dataflow programming uses the same "flow" paradigm.

Dataflow Languages are Naturally Expressed Graphically



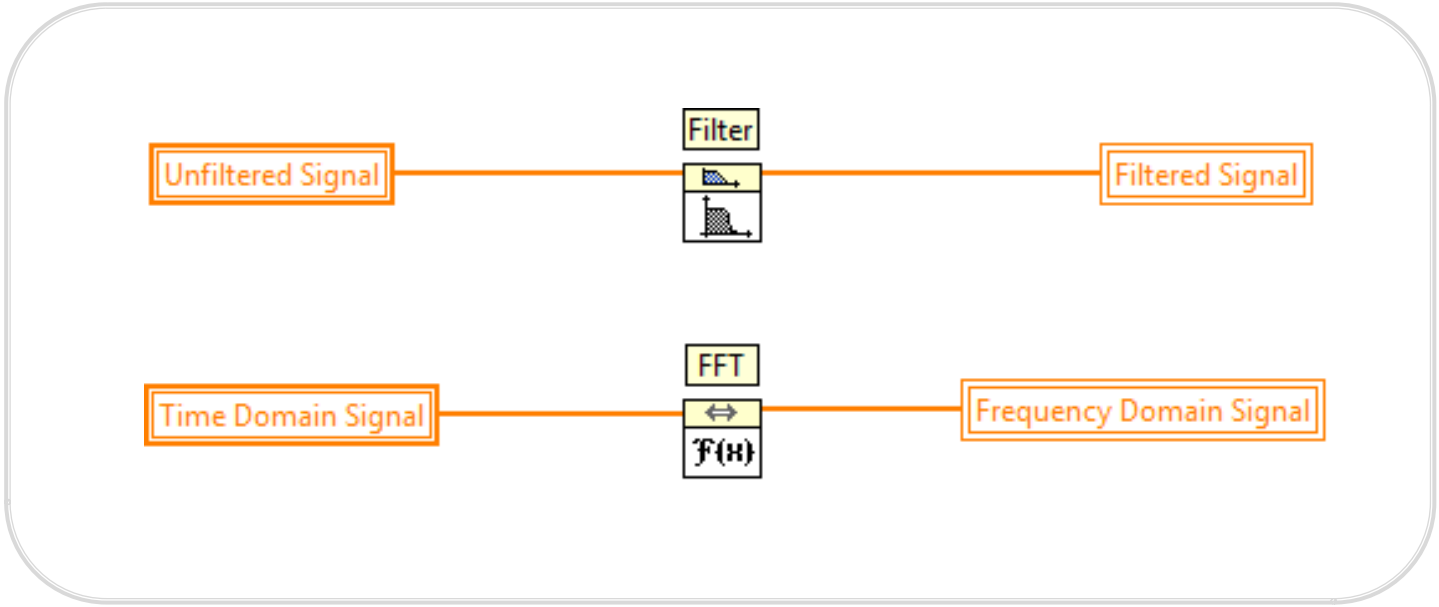
Formula: Result = (A+B*C) / (D-E)

Dataflow Languages Enable Automatic Parallelization



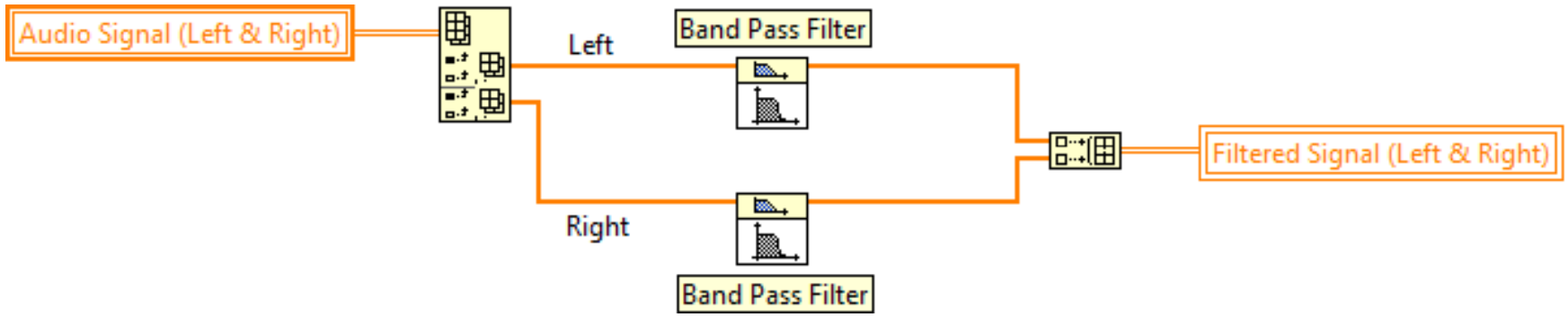
Both the multiply/add and subtract operations can execute at the same time

Dataflow Languages Naturally Express Parallel Applications



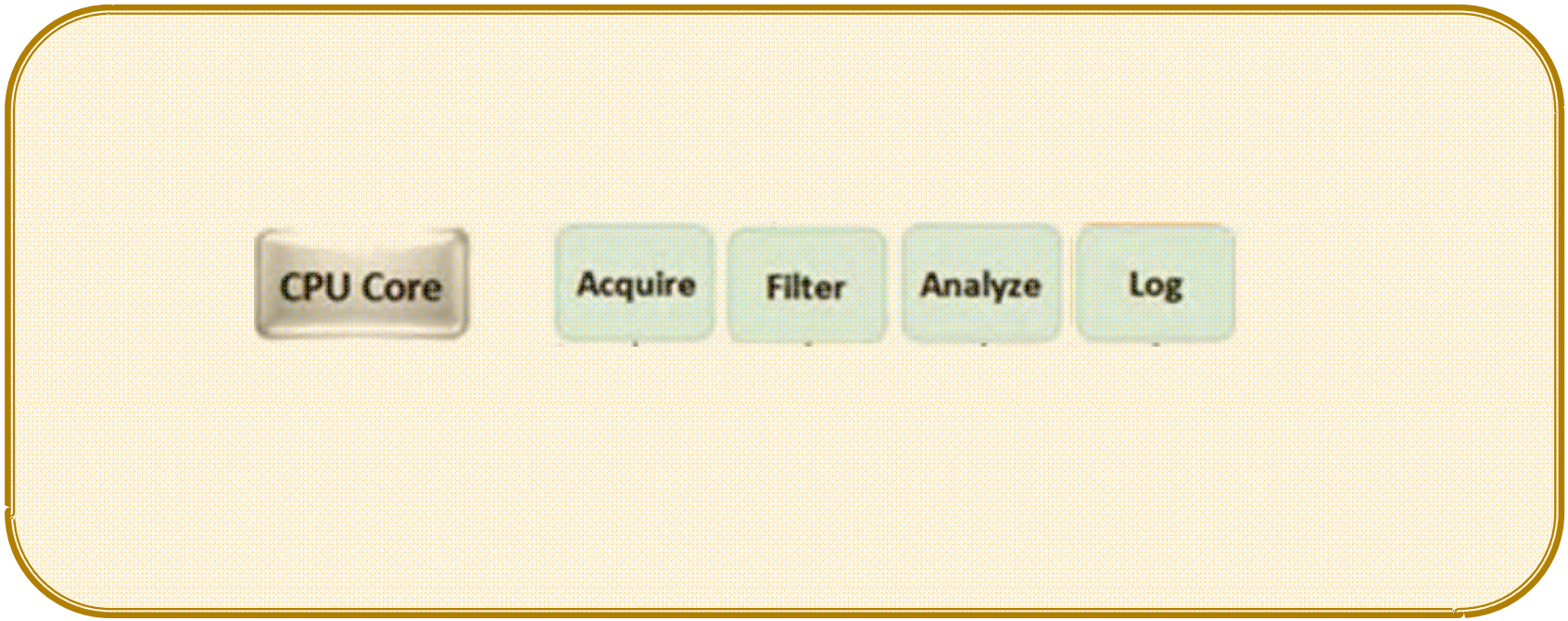
Task Parallelism

Dataflow Languages Naturally Express Parallel Applications

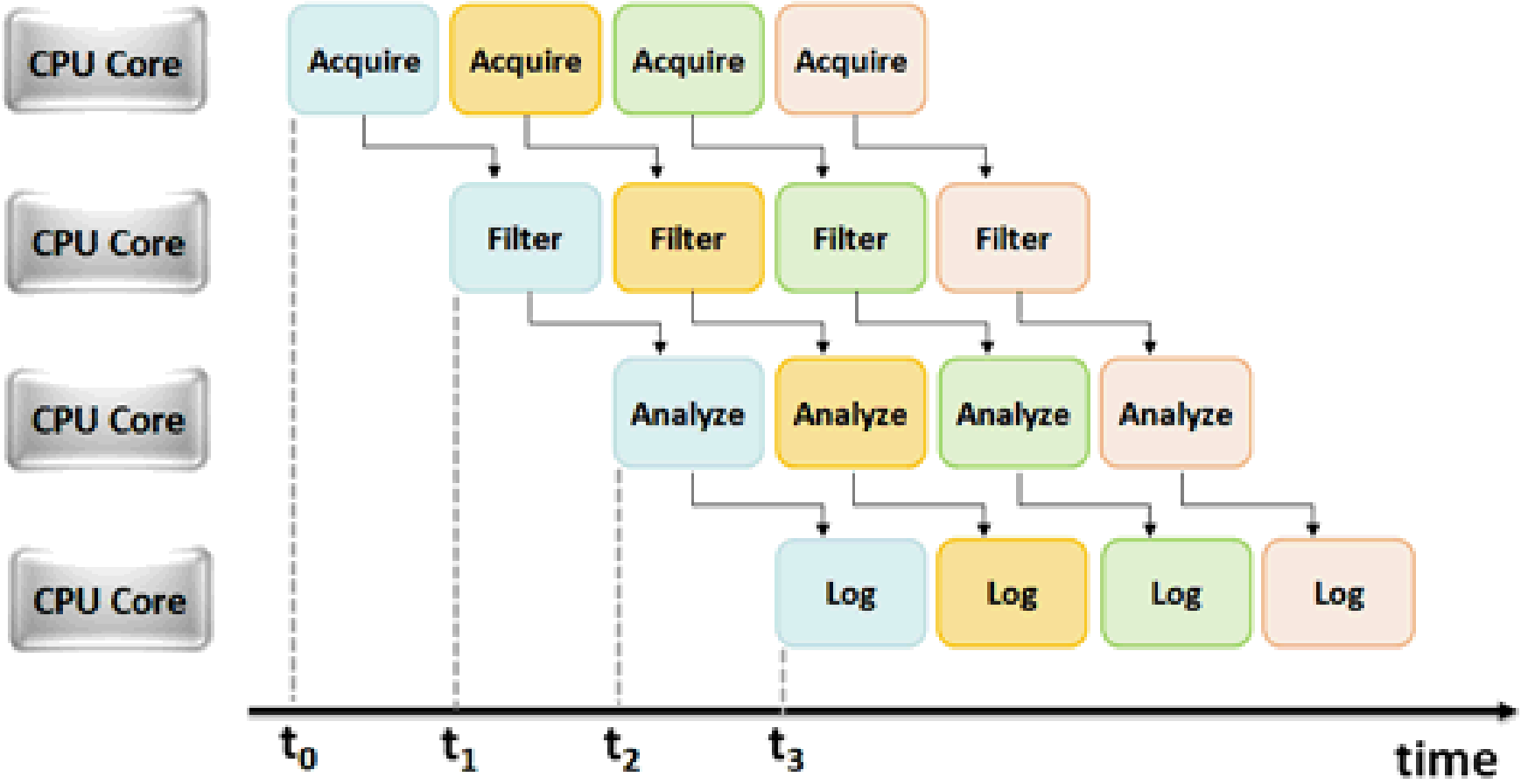


Data Parallelism

Sequential Operations

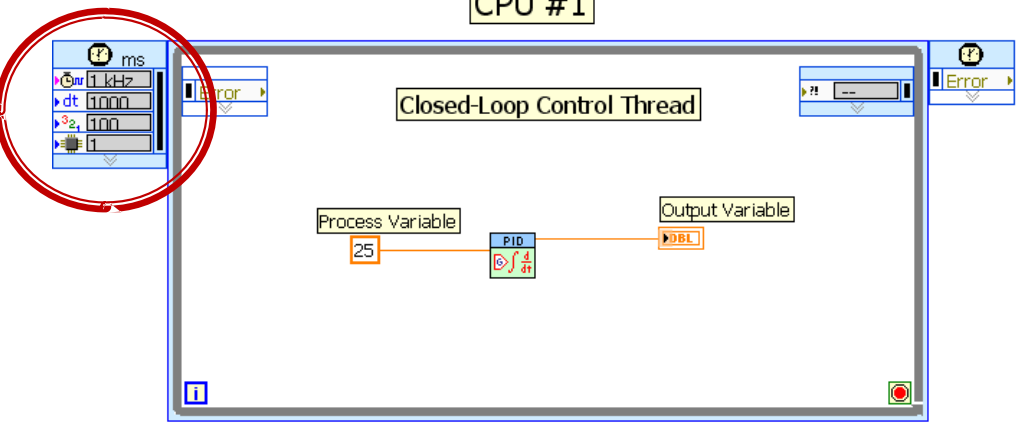


Parallel Computing: Pipelining

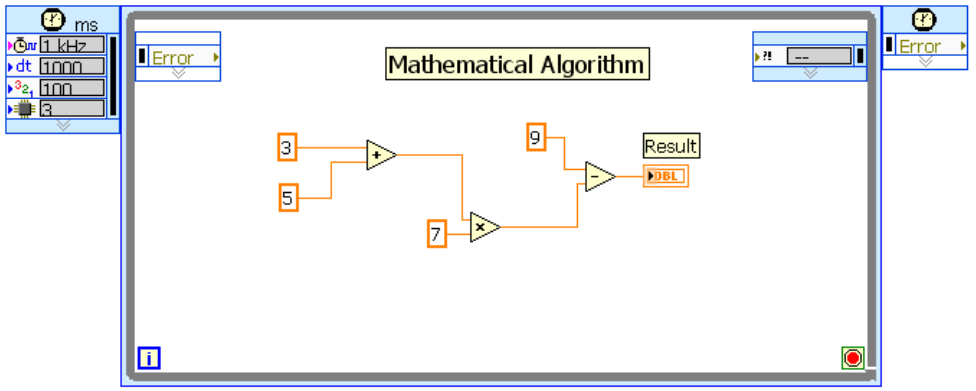


Parallel Operations on Multiple CPUs

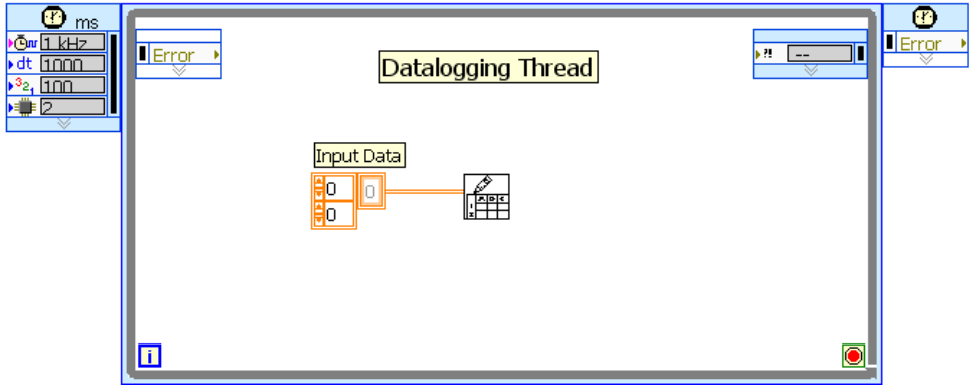
CPU #1



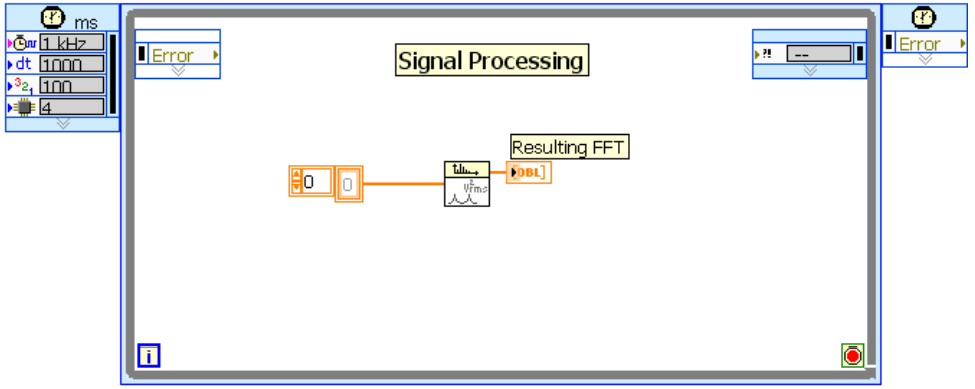
CPU #3



CPU #2



CPU #4

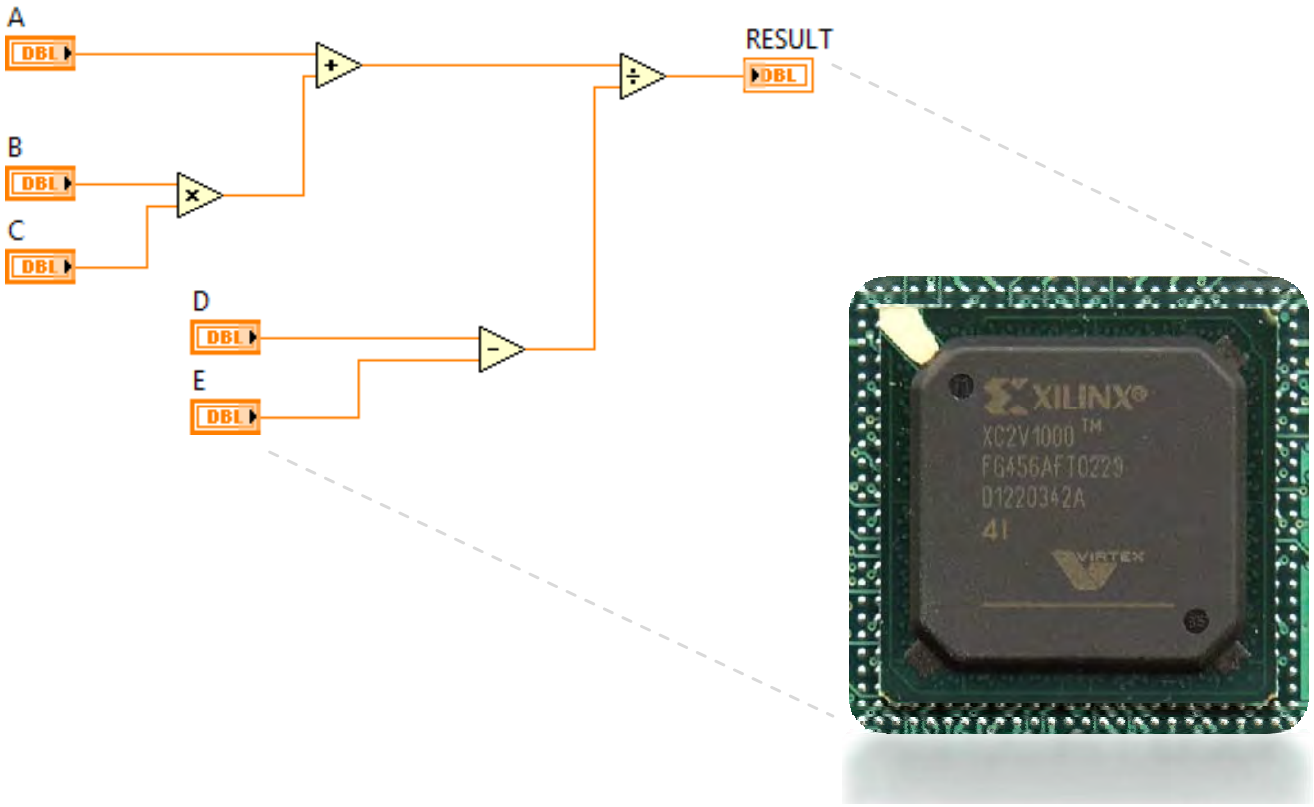


Dataflow Languages are Actively Used in Academia and Industry

- Academic Efforts
  - SISAL (University of Manchester & Colorado State)
  - LUSTRE (University of Victoria)
- Commercial Products and Standards
  - VHDL (based on IEEE standards)
  - National Instruments LabVIEW
  - Agilent VEE
  - Northwoods Software Sanscript
- Many others…

Outside the CPU sphere: Other Parallel Hardware Targets

- Market is demanding smaller, cheaper, faster targets
- FPGAs, DSPs, Embedded Real-time products
- Programmable hardware targets are converging



Advantages and Caveats of Dataflow Languages

Caveats	Advantages
Typically no by-reference data accesses (by-value only)	Can be automatically be mapped to parallel hardware including multicore CPUs
Some overhead due to run-time scheduler (if present)	Naturally expressed graphically; can improve productivity
Different paradigm from imperative languages: requires a learning curve	May reduce the need for multiple development tools

Conclusions

- Increasingly parallel embedded hardware warrants new methods of parallel software development
- Dataflow languages can address some major challenges associated with parallel programming
- Many dataflow languages exist today, and should be considered along with other programming solutions

Email garth.black@ni.com with questions

NI Support at HAFB

- Skilled engineering & developer support. Current work includes:
  - Solar Radiometer System (Embedded Real-time)
  - EFV (Expeditionary Fighting Vehicle)
  - CBATS Test Platform
  - Metrology Lab
- Base Contractor's Badge
- Familiarity and history with base operation
  - Weekly Visits
  - Complimentary - quarterly training sessions

Thank You

References:

- [1] Whiting, P. G., & Pascoe, R. S. (1994). A History of Data-Flow Languages. *IEEE Annals of the History of Computing, Vol. 16, No. 4* , pp. 38-59.
- [2] Lee, E. A., & Messerschmitt, D. G. (1987, September). Synchronous Data Flow. *Proceedings of the IEEE, Vol. 75, No. 9* , pp. 1235-1245.
- [3] Andrade, H. A., & Kovner, S. (1998). Software Synthesis from Dataflow Models for G and LabVIEW. Department of Electrical and Computer Engineering, University of Texas at Austin.
- [4] Arvind, Culler, D. E., & Maa, G. K. (1988). Assessing the Benefits of Fine-grain Parallelism in Dataflow Programs. Labratory for Computer Science, Massachusetts Institute of Technology.
- [5] Johnston, W. M., Hanna, J.R. P., & Millar, R. J. (2004). Advances in Dataflow Programming Languages. *ACM Computing Surveys, Vol. 36, No. 1*, pp. 1-34.
- [6] Lee, B. & Hurson, A.R. (1994). Dataflow Architectures and Multithreading. *Computer, Vol. 27, No. 8*, pp. 27-39.
- **[7]** Asanovic, Krste et al. (December 18, 2006). "The Landscape of Parallel Computing Research: A View from Berkeley" (PDF). University of California, Berkeley. Technical Report No. UCB/EECS-2006-183.

How to contact me?

Garth Black
Field Engineering Manager
N. Utah & E. Idaho
Phone: (801) 447-3343
E-mail: garth.black@ni.com